AFIMB

IRYS Server v2.0.0

PRESENTATION et INSTALLATION

Février 2013

Agence Française pour l'Information Multimodale et la Billettique

Informations sur ce document :

<table>
<tr><td colspan="3">Organisme commanditaire : AFIMB</td></tr>
<tr><td colspan="3">Titre : Présentation et installation de IRYS : serveur SIRI sur base Chouette</td></tr>
<tr><td colspan="3">Sous-titre : Procédure à suivre pour installer et configurer le serveur IRYS.</td></tr>
<tr><td>Organismes auteurs
CITYWAY

Maitre d'ouvrage
DGITM/MTI</td><td>Rédacteurs
Michel ETIENNE</td><td>Participants
Patrick GENDRE
Marc FLORISSON
Michel ETIENNE
Jean SENG
Frédéric SCHETTINI
Laurent DEZOU
Jean-Dominique LECA</td></tr>
<tr><td colspan="3">Résumé :
IRYS est un logiciel libre développé à l'initiative du ministère français chargé des transports (et du développement durable) dans le but de faciliter l'échange de données transport collectif (TC) "temps réel", en s'appuyant pour cela sur SIRI, spécification technique européenne (CEN/TS 15531:2007) en 5 parties qui définit un protocole d'échange (en XML), et plus précisément sur le local agreement français du profil STIF-CERTU.
L'application IRYS est disponible sous la forme d'une application Web destinée à être déployée sur un serveur Web J2EE. Elle implémente les services suivants : Discovery, CheckStatus, StopMonitoring, GeneralMessage.
Ce document présente l'architecture du logiciel IRYS (client et serveur), notamment en association avec le logiciel Chouette pour le démonstrateur visible sur le site www.chouette.mobi
Est présentées ensuite la deux procédure d'installation de l'application, à partir d'une distribution de sources, ou à partir d'une archive WAR exécutable.</td></tr>
<tr><td colspan="2">Mots clés :
profil d'échange Neptune, SIRI, information multimodale, application Chouette, manuel d'installation, postgreSQL, Tomcat, java</td><td colspan="1">Diffusion :
publique (licence Creative Commons CC-by-nd)</td></tr>
<tr><td>Nombre de pages :
18 pages</td><td>Date
Février 2013</td><td>Confidentialité :
Non</td><td>Bibliographie :
Oui</td></tr>
</table>

Historique des versions / révisions :

Historique des versions / révisions			
Version/ révision	Date d'application	Description des changements	Auteur
2.0.0	10/10/2012	Première version	Michel Etienne
	05/02/13	Fusion avec document DRYADE de présentation de la Démo SIRI de mars 2011	Patrick Gendre

CC BY ND

Sommaire

1. Présentation d'IRYS, application client/serveur SIRI 5

1.1 Rappel sur SIRI 5

1.2 Architecture d'IRYS, articulation avec CHOUETTE 6

1.3 Le démonstrateur SIRI 9

2. Installation 10

2.1 Pré-Requis 10

2.2 Récupération du projet IRYS 10

2.3 Base de données, ajout des tables temps-réel 10

2.4 Installation à partir des sources 11

2.5 Installation à partir de l'archive .war 11

2.6 Exploitation et maintenance 18

cc BY ND

1. Présentation d'IRYS, application client/serveur SIRI

1.1 Rappel sur SIRI

Afin de fournir aux transporteurs et aux industriels un cadre normalisé pour l'échange de données concernant l'information temps réel, le CEN TC278 -WG3-SG7 a décidé de lancer le projet **SIRI** (**S**ervice **I**nterface for **R**ealtime **I**nformation) dès 2004.

SIRI est une norme (TS) d'échange d'information inter Systèmes orientée vers les Systèmes d'Aide à l'Exploitation et les Systèmes d'Information Voyageur. SIRI a été développé pour être compatible avec les normes et standards pré-existants que sont TransModel (modèle de données conceptuel pour le transport en commun), IFOPT (description des lieux d'arrêt) ainsi que DATEX et TPEG. Dans le contexte Français, SIRI est un complément « naturel » à NEPTUNE (profil d'échange de l'offre théorique) : un profil dit « CERTU-STIF » a été publié http://www.predim.org/spip.php?article3387.

Une version 2 de SIRI a été élaborée et soumise au vote en 2012.

Pour en savoir plus ; http://www.normes-donnees-tc.org/spip.php?rubrique19

SIRI se présente sous la forme d'un ensemble de services :

- Horaires planifiés (***Production Timetable Service***) pour la mise à jour des horaires théoriques, des missions et des itinéraires "publiés" (mise à jour en amont de la journée d'exploitation considérée)
- Horaires calculés (***Estimated Timetable Service***) : horaires estimés pendant la course, et mise à jour des missions et des itinéraires.
- Horaires à l'arrêt (***Stop Timetable Service***) : Information horaire à l'arrêt (théorique, planifié et commandé: mise à jour en amont de la journée d'exploitation considérée). Ce service n'a pas été retenu dans le Profil IDF-CERTU.
- Passage à l'arrêt (***Stop Monitoring Service***): information sur les prochains bus passant à un arrêt (calculé uniquement).
- Supervision des véhicules (***Vehicle Monitoring Service***): localisation d'un véhicule au cours d'une course.
- Gestion des correspondances planifiées (***Connection Timetable Service***) : mise à jour des informations sur les correspondances (par rapport aux informations théoriques). Ce service n'a pas été retenu dans le Profil IDF-CERTU.
- Gestion des correspondances calculées (***Connection Monitoring Service***) : information et gestion des correspondances pendant l'exploitation.
- Messagerie (***General Messaging Service***) : messagerie « générique »
- Supervision des équipements (***Facility Monitoring***) : information sur l'état des équipements, avec un accent particulier sur les besoins PMR et les besoins spécifiques
- Gestion des événements (***Situation Exchange***): modèle structuré d'incident et description détaillée des conséquences.

SIRI est conçu comme une boite à outils qui permet une implémentation progressive et peut s'adapter à des systèmes existants pour en diffuser les informations dans un format normalisé.

IRYS implémente les services minimums obligatoires dans le cadre du Profil IDF-CERTU et peut être complétée de l'ensemble des services SIRI en fonction des besoins des projets.

CC BY ND

1.2 Architecture d'IRYS, articulation avec CHOUETTE

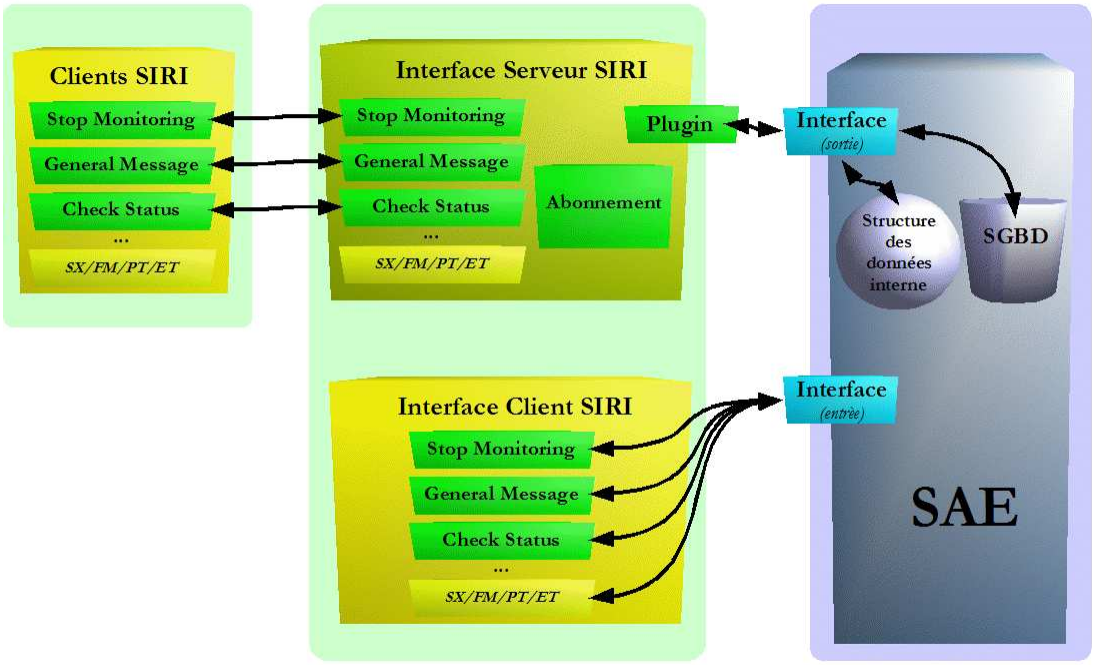
IRYS facilite la mise en conformité SIRI de systèmes existants ou dans le contexte de nouveaux développements.

IRYS est un frontal non intrusif de diffusion de données qui échange avec un autre client / serveur SIRI. CHOUETTE peut jouer le rôle de frontal SAE : le serveur SIRI interroge alors en fait Chouette via un composant (plug-in) qui répond en SIRI aux requêtes du serveur.

L'application IRYS est disponible en 2 versions :

- IRYS (générique) : correspondant à la figure ci-dessous

- IRYS-CHOUETTE : correspondant à la figure plus bas présentation le démonstrateur, où une base de données CHOUETTE est intégrée à l'application et complétée par des données temps réel.



IRYS est modulaire et permet une mise en place de SIRI progressive. L'application prend en compte nativement les services :

1. StopMonitoring : information temps-réel au passage à un arrêt
2. GeneralMessage : information textuelle temps-réel
3. Discovery : information simplifiée sur la topologie du réseau (liste des arrêts et des lignes)
4. CheckStatus : demande d'état du système

L'architecture permet, suivant les fonctionnalités attendues de proposer un service en mode requête direct ou en mode abonnement. IRYS propose la mise à disposition de clients SIRI (mode requête ou abonnement, là encore) :

BY ND

- Comme base pour mettre en place des services d'information
- Pour alimenter un SAE (le SAE, naturellement fournisseur, peut aussi être client pour proposer une information multi-exploitants, ou proposer des services de régulation exploitant les informations d'autres transporteurs).
- Pour alimenter un pilote d'afficheur
- Pour réaliser des tests, des validations ou encore des opérations de qualification

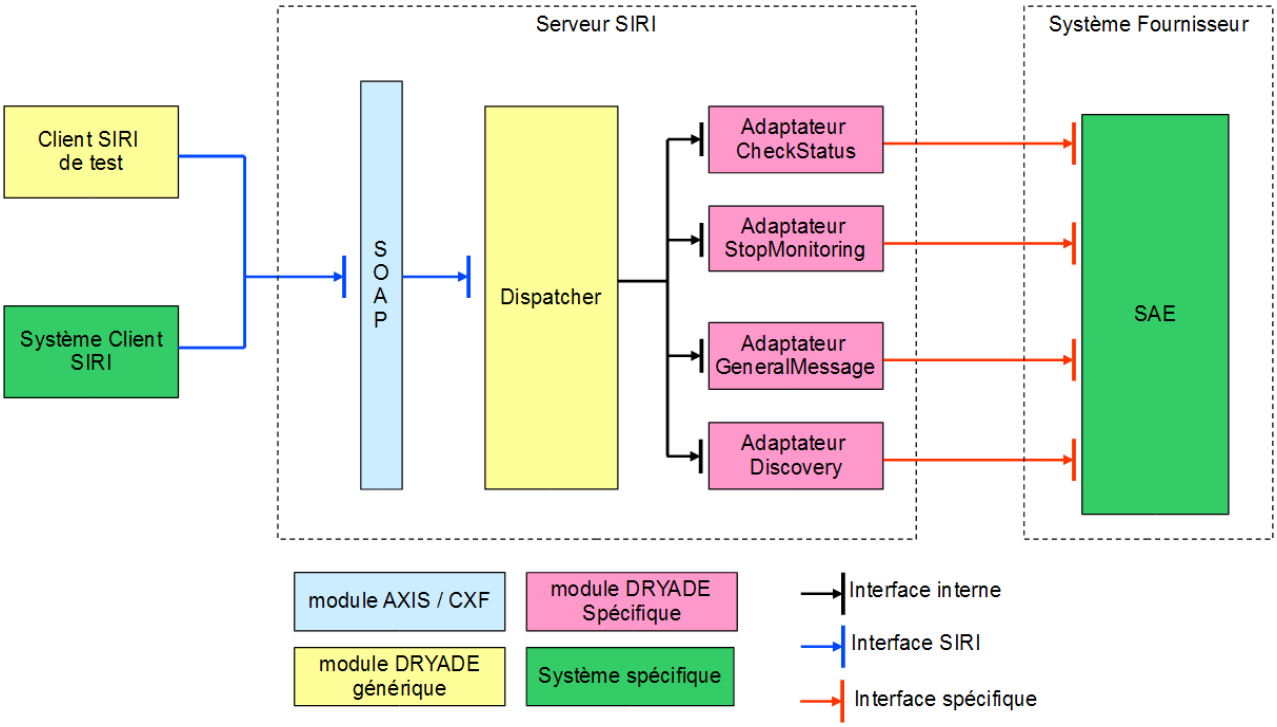
Le client est proposé en mode ligne de commande (avec possibilité de « scripting ») en en mode client Web pour une mise en œuvre simple et interactive.

La solution est multi plates-formes (Linux, Windows, etc.) et est conforme au profil défini par le STIF et le CERTU.

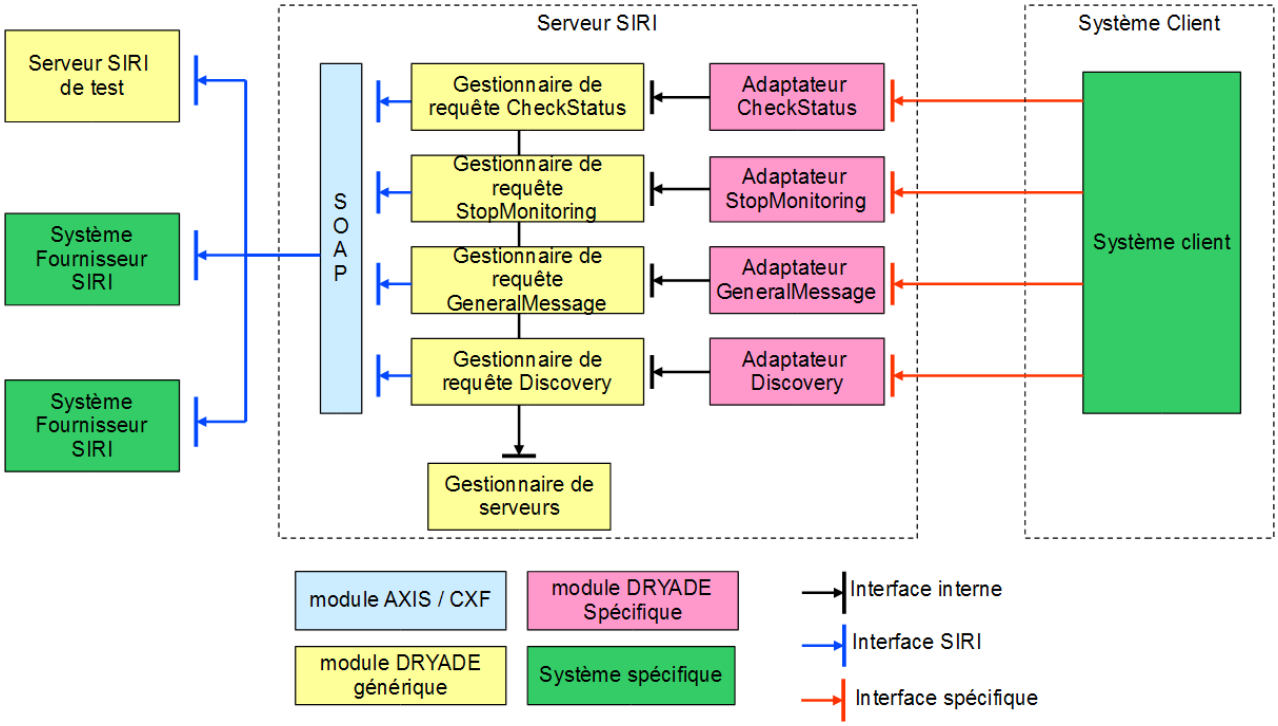
Les figures ci-dessous proposent une vue synoptique de l'architecture du serveur et du client SIRI. Parmi ses principales caractéristiques, on peut noter :

- l'implémentation des méthodes des services WEB,
- le contrôle de validité des requêtes / réponses par rapport aux exigences SIRI et au profil IDF/CERTU,
- l'implémentation de « bouchons » permettant un premier niveau de test avant même la connexion au service fournisseur, et retournant une erreur sur les méthodes non implémentées,
- la disponibilité d'un mécanisme de plug-in pour se connecter à un SAE (ou tout autre système fournisseur),
- la disponibilité de l'accès aux services en mode requête simple et en mode abonnement,
- un programme client de test permettant de valider les flux de données en retour des requêtes SIRI ainsi que les performances,
- un programme serveur de test permettant de simuler des flux SIRI ; ce programme utilise une base normalisée au format Neptune,
- l'utilisation du framework SOAP : Apache Axis 2 avec une possibilité de basculement sur Apache CXF,
- l'utilisation du modèle Apache XmlBeans.

CC BY ND



Architecture Serveur SIRI



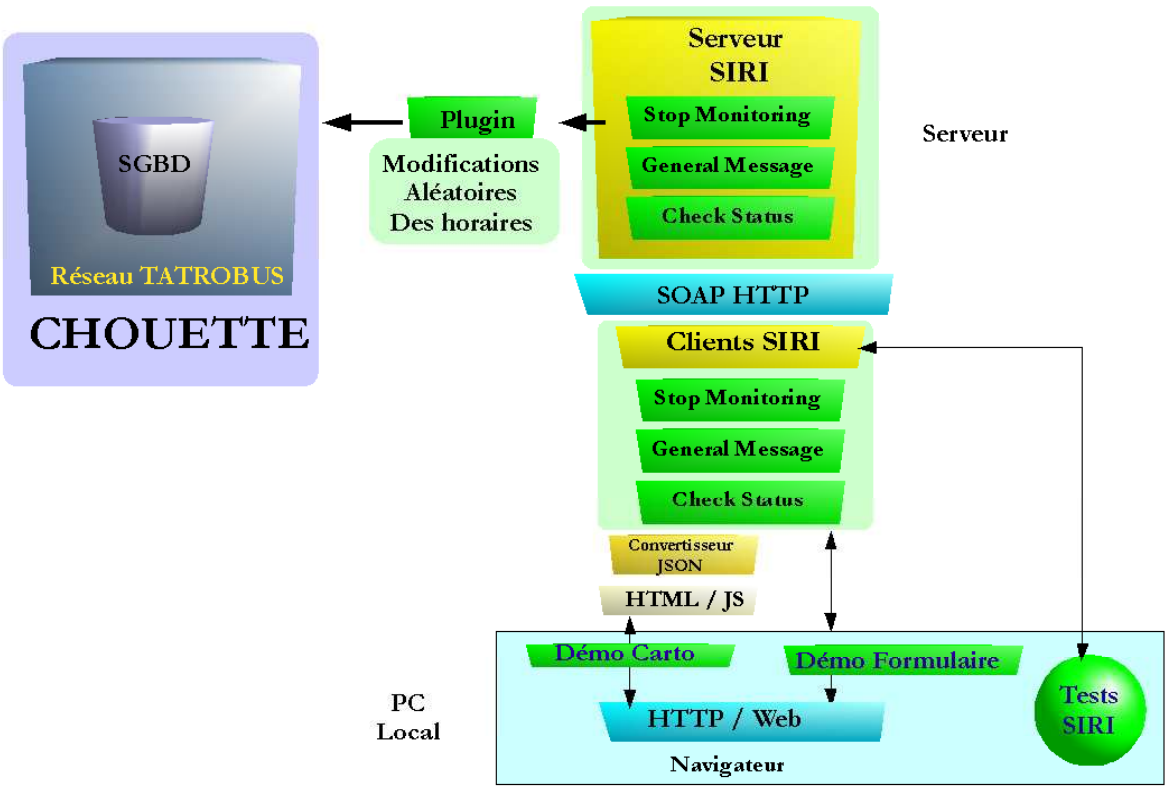
Architecture Client SIRI

CC BY ND

1.3 Le démonstrateur SIRI

Le démonstrateur est configuré avec les éléments suivants :

- Une base Chouette enrichie d'un réseau de simulation TATROBUS,
- Un serveur SIRI sous la forme d'un exécutable,
- Un Client SIRI pour dialoguer directement avec le serveur SIRI,
- Une application Démo en contexte cartographique permettant de consulter le réseau de test TATROBUS,
- Une application Démo en contexte Web
- Une application Démo via un formulaire Http
- Un composant logiciel qui modifie aléatoirement les horaires théoriques en alimentation de Chouette,
- Le client de tests pour mettre en oeuvre les services StopMonitoring, GeneralMessage et CheckStatus.



CC BY ND

2. Installation

2.1 Pré-Requis

L'application serveur IRYS-CHOUETTE est basée sur l'architecture de Chouette, se référer au manuel d'installation de Chouette pour les pré-requis (JRA, Tomcat, PostgreSQL, Post-GIS, ainsi que Maven2, JDK, git pour installation à partir des sources).

2.2 Récupération du projet IRYS

2.2.1 Sources

Les sources sont disponibles sur le site internet www.chouette.mobi dans la rubrique SIRI / Téléchargements, qui vous invite à nous laissez vos coordonnées avant d'obtenir les sources, afin de mieux connaître nos utilisateurs: merci d'avance de nous signaler qui vous êtes avant d'installer l'application !

L'archive contenant les sources est accessible sur github aux adresses :

http://github.com/dryade/IRYS : code java de l'application générique

http://github.com/dryade/IRYS-CHOUETTE: code java de l'application associée à une base de données Chouette

La présente documentation détaillera l'installation de IRYS-CHOUETTE.

2.2.2 Archive war

L'archive contenant le fichier war de Irys-Chouette prêt à déployer et du client est disponible sur le site internet www.chouette.mobi dans la rubrique SIRI / "Téléchargements".

2.3 Base de données, ajout des tables temps-réel

IRYS-Chouette utilise la base Chouette V2.0 , se référer au manuel d'installation associé

La base de données Chouette n'est pas suffisante pour gérer le temps-réel, un complément est donc nécessaire.

2.3.1 Paramètres associés à la base

Le tableau suivant récapitule les paramètres relatifs à la base. Tous ces paramètres peuvent être adaptés, il s'agit seulement de rester cohérent sur leur valeur tout au long de la procédure d'installation.

Paramètre	Valeur proposée
la machine d'hébergement est la machine locale	localhost
le nom de la base	chouette2
le nom du compte applicatif et propriétaire du schéma	chouette
le mot de passe du compte applicatif	chouette

CC BY ND

2.3.2 Création du schéma "siri"

Le script «siri.sql » est disponible dans l'archive irys_server_2.0.0-distribution.zip

```
psql -U <propriétaire> -W -h localhost -f siri.sql -v schemaname=<schema> -v username=<propriétaire> -d <base>
```

Le nom du schéma doit être en MINUSCULES

2.4 Installation à partir des sources

Voir le manuel d'installation CHOUETTE, la procédure est similaire pour IRYS (avec Maven).

2.5 Installation à partir de l'archive .war

Ce chapitre explique les étapes de déploiement de l'archive WAR.

Cette archive est générée avec les paramètres d'application exposés ci-dessous.

2.5.1 Installation du serveur

Le serveur SIRI sera installé en tant que web service sous tomcat :

1. créer le répertoire de configuration: C:\Program Files\Irys\Server\conf\<nom_webapp> ou /etc/irys/server.conf/<nom_webapp> (selon système)
2. décompacter l'archive irys_server_2.0.0-distribution.zip dans un répertoire temporaire
3. copier le fichier log4j.properties dans le répertoire de configuration
4. copier le fichier irys.properties dans le répertoire de configuration
5. configurer le serveur SIRI : (les paramètres à régler sont en **bleu** dans les fichiers suivants)
6. copier le fichier irys_server.war dans le répertoire <chemin d'installation de tomcat>/webapps en le **renommant** <nom de webapp>.war (ce nom sera visible dans l'URL publique)
7. verifier qu'un dossier <nom de webapp> se crée à coté, dans le cas contraire, relancer le service Tomcat.

paramétrage de irys.properties

```
#Fichier de configuration pour les services SIRI IDF:

## Parametres generaux
# reference operateur rendue par d\efaut
siri.operator=SIRI:Company:dryade:LOC
# reference producteur de donnees (premier item des references SIRI)
siri.producerRef=<Nom du système émetteur des données>
# adresse publique du serveur soap
#siri.serverURL=http://demo-siri.dryadebox.net/SiriServer
siri.serverURL=<url publique du serveur>
```


CC BY ND

```
# activation de la validation de la requete complete
siri.validation.request=true
# activation de la validation de la reponse
siri.validation.response=false

## Gestion des identifiants du referentiel (par defaut iden-
tifiant local); si false alors referentiel commun
## ces trois items sont a passer a false pour un reseau Ile-
De-France
siri.idWithLoc.Line=true
siri.idWithLoc.Network=true
siri.idWithLoc.Company=true
siri.idWithLoc.JourneyPattern=true
siri.idWithLoc.VehicleJourney=true
siri.idWithLoc.Route=true
siri.idWithLoc.StopPoint.SP=true
siri.idWithLoc.StopPoint.BP=true
siri.idWithLoc.StopPoint.QUAY=true
siri.idWithLoc.StopPoint.SPOR=true
siri.allowedStopPointTypes=SPOR,BP,Q,SP


## Parametres StopMonitoring
# version de compatibilite SIRI assuree par le service
siri.stopmonitoring.version=1.3
# presence des stat dans les logs
siri.stopmonitoring.stat=true
# seuil en seconde pour déclarer un bus en avance
siri.stopmonitoring.earlyGap=60
# seuil en seconde pour déclarer un bus en retard
siri.stopmonitoring.delayedGap=180

# Parametres GeneralMessage
# version de compatibilite SIRI assuree par le service
siri.generalmessage.version=1.3
# reference de format de message
siri.generalmessage.formatRef=SIRI-IDF
# liste des canaux de messages
si-
ri.generalmessage.channels=Perturbation,Information,Commercia
l
# liste des acronymes de langue
siri.generalmessage.langs = FR,EN
# langue par defaut si non precisee dans les requetes
siri.generalmessage.defaultLang = FR
# specifie si les infochannels sont passes en texte (false)
ou codes (true)
siri.generalmessage.encoded=false


# Parametres Discovery
# version de compatibilite SIRI assuree par le service
siri.discovery.version = 1.3

## Parametres d'implementation metier
```


cc BY ND

```
siri.wsdl.version=1.3
# nom de la classe implementant le service stopMonitoring
siri.StopMonitoringService=chouetteStopMonitoringService
# nom de la classe implementant le service generalMessage
siri.GeneralMessageService=generalMessageService
# nom de la classe implementant le service checkStatus
siri.CheckStatus=checkStatus
# nom de la classe implementant le service DataSupply
siri.DataSupplyService=
# nom de la classe implementant le service Subscription
siri.SubscriptionService=subscriptionService
# nom de la classe implementant le service Capabilities
siri.CapabilitiesService=
# nom de la classe implementant le service ConnectionMonito-
ring
siri.ConnectionMonitoringService=
# nom de la classe implementant le service ConnectionTimeta-
ble
siri.ConnectionTimetableService=
# nom de la classe implementant le service EstimatedTimetable
siri.EstimatedTimetableService=
# nom de la classe implementant le service FacilityMonitoring
siri.FacilityMonitoringService=
# nom de la classe implementant le service ProductionTimeta-
ble
siri.ProductionTimetableService=
# nom de la classe implementant le service StopTimetable
siri.StopTimetableService=
# nom de la classe implementant le service SituationExchange
siri.SituationExchangeService=
# nom de la classe implementant le service VehicleMonitoring
siri.VehicleMonitoringService=
# nom de la classe implementant le service Discovery
siri.DiscoveryService=chouetteDiscoveryService

# chouette
database.name = <nom de la base (exemple chouette)>
database.schema = chouette
database.showsql = false
hibernate.username = chouette
hibernate.password = chouette
jdbc.url=jdbc:postgresql://localhost:5432/${database.name}
jdbc.driver=org.postgresql.Driver
database.hbm2ddl.auto=update
system.code=<préfixe des identifiants Neptune dans la base
chouette>

# reload referential cache on Network's VersionDate change
siri.referential.scanNetworkVersionDate=true
siri.referential.scanPeriod=30

# siri database
siri.jdbc.driver=org.postgresql.Driver
siri.jdbc.url=jdbc:postgresql://localhost:5432/<nom de la
base (exemple chouette)>
```


cc BY ND

```
siri.jdbc.username= chouette
siri.jdbc.password= chouette
siri.schema=siri
```

paramétrage de log4j.properties

```
#
=============================================================
=============#
#                     Definition des cibles pour les traces
#
=============================================================
=============#
log4j.rootCategory=ERROR, R

#
=============================================================
=============#
#                     Parametrages pour la cible: stdout
#
=============================================================
=============#
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%p
%X{username} %C{1} - %m\n

#
=============================================================
=============#
#                     Parametrages pour la cible: R
#
=============================================================
=============#
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.File=<chemin absolu des fichiers de log>
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{ISO8601} %-5p
%C{1} %X{username} - %m\n
# Taille maximmum du fichier de trace
log4j.appender.R.MaxFileSize=1000KB
# Total de fichiers d'archive
log4j.appender.R.MaxBackupIndex=5

#
=============================================================
=============#
#          Niveaux des traces du composant Hibernate
#
=============================================================
=============#
log4j.logger.org.hibernate=ERROR
log4j.logger.org.hibernate.engine.StatefulPersistenceContext=
ERROR
```


CC BY ND

```
log4j.logger.org.hibernate.tool.hbm2ddl=WARN

#
============================================================
=============#
#          Niveaux des traces du composant Spring
#
============================================================
=============#
log4j.logger.org.springframework=WARN
log4j.logger.org.springframework.transaction=WARN
log4j.logger.org.springframework.orm.hibernate3=WARN
log4j.logger.org.springframework.ws=INFO


#
============================================================
=============#
#          Niveaux des traces de l'application irys
#
============================================================
=============#
log4j.logger.irys.siri=INFO
```

2.5.2 Installation du générateur de données Temps Réel (mode démo)

Le générateur de données Temps Réel génère des courses TR statiques pour la journée d'exploitation en exploitant les calendriers de Chouette.

Ces courses sont légèrement décalées par rapport au théorique mais n'évoluent pas dans la journée.

Des données de type GeneralMessaging sont aussi générées aléatoirement à partir de modèles définis dans le jar de l'application (fichier irysContext.xml)

Ces messages sont éventuellement affectés a des lignes et des arrêts du réseau défini dans la base Chouette

Le générateur de données TR est livré dans l'archive :

irys_realtime_simulator_2.0.0.zip

Le décompresser dans un répertoire d'installation au choix. Il contient :

Fichier	description
irys-chouette-realtime-simulator-2.0.0.jar	Application simulateur
lib	Librairies applicatives du simulateur
simulator.bat	Lanceur en mode windows
simulator.sh	Lanceur en mode unix
irys-client.properties	Paramètres SIRI

cc BY ND

Fichier	description
log4j.properties	Paramètres de log

paramétrage de irys-client.properties

```
##Fichier de configuration pour les services SIRI IDF:

#------------------------
#  CLIENT
#------------------------

# chouette database access
database.name = <nom de la base (exemple chouette)>
database.schema = chouette
database.showsql = false
hibernate.username = chouette
hibernate.password = chouette
jdbc.url=jdbc:postgresql://localhost:5432/<nom de la base (exemple chouette)>
jdbc.driver=org.postgresql.Driver
database.hbm2ddl.auto=update
system.code=<préfixe des identifiants Neptune dans la base chouette>

# reload referential cache on Network's VersionDate change
siri.referential.scanNetworkVersionDate=false
siri.referential.scanPeriod=30

# SIRI database access
realtime.jdbc.url=jdbc:postgresql://localhost:5432/<nom de la base (exemple chouette)>
realtime.jdbc.username=chouette
realtime.jdbc.password=chouette
realtime.hbm2ddl.auto=update
realtime.showsql=false
realtime.schema=siri
```

paramétrage de log4j.properties

```
#
=============================================================
=============#
#                     Definition des cibles pour les traces
#
=============================================================
=============#
```


cc BY ND

```
log4j.rootCategory=ERROR, stdout, R

#
=============================================================
=============#
#                     Parametrages pour la cible: stdout
#
=============================================================
=============#
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ISO8601} %-
5p %C{1} %X{username} - %m\n

#
=============================================================
=============#
#                     Parametrages pour la cible: R
#
=============================================================
=============#
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.File=<chemin absolu des fichiers de log>
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{ISO8601} %-5p
%C{1} %X{username} - %m\n
# Taille maximmum du fichier de trace
log4j.appender.R.MaxFileSize=1000KB
# Total de fichiers d'archive
log4j.appender.R.MaxBackupIndex=5

#
=============================================================
=============#
#          Niveaux des traces du composant Hibernate
#
=============================================================
=============#
log4j.logger.org.hibernate=ERROR
log4j.logger.org.hibernate.engine.StatefulPersistenceContext=
ERROR
log4j.logger.org.hibernate.tool.hbm2ddl=WARN

#
=============================================================
=============#
#          Niveaux des traces du composant Spring
#
```


cc BY ND

```
#============================================================
=============#
log4j.logger.org.springframework=WARN
log4j.logger.org.springframework.transaction=WARN
log4j.logger.org.springframework.orm.hibernate3=WARN

#
#============================================================
=============#
#          Niveaux des traces de l'application irys
#
#============================================================
=============#
log4j.logger.irys.siri=INFO
```

Le simulateur sera lancé tous les jours de bon matin en paramétrant la crontab sous linux ou une tâche planifiée sous Windows

2.6 Exploitation et maintenance

Les opérations d'import XML et CSV modifient des volumes importants de données.

Il devient important de mettre à jour les index et les statistiques des données en base. L'opération « vacuum analyse » doit donc être lancée de façon régulière.

La version 8.3 de PostgreSQL permet de réaliser cette tache dans un traitement à part (démon auto-vacuum cf http://www.postgresql.org/docs/8.3/static/maintenance.html)

Le travail de l'admin BDD peut être très variable selon la plateforme et la version de PostgreSQL utilisées.

cc BY ND